A-LEVEL COMPUTER SCIENCE • EXERCISE SHEET

10.3 Files

Name: _____ Class: _____ Date: _____ **Total: 16 marks**

Objective

Build the skills to answer exam questions on **text files** 文本文件—why files are needed, and reading and writing them in pseudocode.

You must be able to:

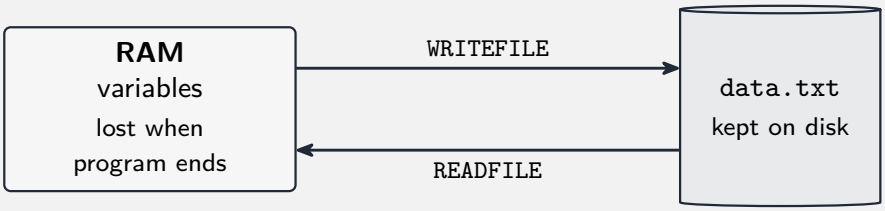
- explain **why** a program uses files
- write pseudocode to **read** a text file line by line using `EOF`
- write pseudocode to **write** or **append** lines to a text file

1 Worked examples

Study these first. Each one shows the method for a question type used later —follow the steps and you can do the Practice and Exam-style questions yourself.

Why files

Variables live in **RAM** and are lost when the program ends. A **file** on disk is **kept** between runs —so it stores data permanently (high scores, records) and lets programs share data.



Reading a text file

Open it, read each line until **end of file**, then **close** it:

```
OPENFILE "data.txt" FOR READ
WHILE NOT EOF("data.txt") DO
    READFILE "data.txt", LineString
    OUTPUT LineString
ENDWHILE
CLOSEFILE "data.txt"
```

`EOF` returns `TRUE` when there is nothing left to read —test it **before** each `READFILE`.

Writing and appending

`FOR WRITE` creates a new file (or **overwrites** an old one); `FOR APPEND` keeps the existing lines and adds after them:

```
OPENFILE "log.txt" FOR WRITE
FOR i ← 1 TO 100
    WRITEFILE "log.txt", "Event " & i
NEXT i
CLOSEFILE "log.txt"
```

Always **close** a file —otherwise buffered writes can be lost and the file may stay locked.

2 Practice

Now apply the methods above.

2.1 State **one** reason a program needs to use a file rather than only variables. [1]

2.2 State when the function `EOF` returns `TRUE`. [1]

2.3 Write pseudocode to **open** `"scores.txt"` for reading and then **close** it. [2]

2.4 State **one** reason you must always **close** a file. [1]

3 Exam-style questions

3.1 Write pseudocode to read **every line** of `"names.txt"` and output each line. [4]

3.2 Write pseudocode to write the numbers **1 to 50**, one per line, to a new file `"nums.txt"`.[3]

3.3 A file `"members.txt"` holds one name per line. Write pseudocode to **count** how many names are in the file and output the total. [4]

4 Go further

You are now ready for the real exam questions on this subtopic. Open the **10.3 Files** past-paper sheet in the Library, or try these in **Practice mode**:

- 9618/21 J24 —Q8 (text-file processing)
- 9618/22 N25 —Q3 (reading and writing files)
- 9618/21 N25 —Q4 (files and EOF)
- 9618/23 J25 —Q4 (file handling)

Solutions

2.1 Any one: data is **kept** after the program ends; it stores large amounts permanently; it lets programs share data / restart from a saved state.

2.2 When the **end of the file** has been reached —there is nothing left to read.

2.3

```
OPENFILE "scores.txt" FOR READ
CLOSEFILE "scores.txt"
```

2.4 Any one: so **buffered writes are saved**; so the file is not left **locked** for other programs.

3.1

```
OPENFILE "names.txt" FOR READ
WHILE NOT EOF("names.txt") DO
    READFILE "names.txt", Line
    OUTPUT Line
ENDWHILE
CLOSEFILE "names.txt"
```

3.2

```
OPENFILE "nums.txt" FOR WRITE
FOR i ← 1 TO 50
    WRITEFILE "nums.txt", i
NEXT i
CLOSEFILE "nums.txt"
```

3.3

```
Count ← 0
OPENFILE "members.txt" FOR READ
WHILE NOT EOF("members.txt") DO
    READFILE "members.txt", Name
    Count ← Count + 1
ENDWHILE
CLOSEFILE "members.txt"
OUTPUT Count
```